Logic and computer science

Sissejuhatus infotehnoloogiasse

Pawel Sobocinski, 1 December 2023

Logical literacy

“Supporters of Putin oppose sending weapons to Ukraine. My opponent is against sending weapons to Ukraine. Therefore he supports Putin!”

Analysing the logical content

- “My opponent is against sending weapons to Ukraine. Therefore he supports Putin!”
- W = “sending weapons to Ukraine”
- SP = “supporting Putin”
- ¬W → SP ?
- It is easy to argue that W → ¬ SP. Is this the same as ¬W → SP? What about SP → ¬W?

Syntax and Semantics

- “¬W → SP” is a sentence in a formal language called **Propositional Logic**
- A big part of computer science is understanding formal languages
  - regular expressions
  - database queries (SQL, datalog, …)
  - programming languages
- The **syntax** is the rules of forming correct sentences
- The **semantics** is the meaning of sentences
- What is the meaning of “¬W → SP”?

Semantics of propositional logic

- the syntactic entities W and SP are called **propositional variables**
- the semantics of a propositional variable is either **T**rue or **F**alse.
- the assignment of truth values to propositional variables gives us a possible world to reason in
- what is the meaning of →?
- X → Y is an operation that takes two arguments. We can give its semantics by listing all of the cases.

X	**Y**	**X→Y**
T	T	T
T	F	F
F	T	T
F	F	T

Compositional semantics

- How to work out the semantics of a sentence like “¬W → SP”?
- The semantics is given **compositionally**. The meaning of whole is the sum of its parts.
- Suppose that we are in a universe where [[W]] = F and [[SP]] = T.
- Then [[¬W → SP]] = [[¬W]]→[[SP]] = ¬[[W]] → [[SP]] = ¬F → T = T → T = T

Axioms and Theories

- A propositional theory consists of some propositional formulas that we accept as true
- Let us consider the theory with a single axiom: SP → ¬W
- A **model** of this theory is any truth assignment to SP and W that makes SP→¬W evaluate to true
- Now we can return to the original question: is it the case that if SP → ¬W then ¬W → SP?

Counterexamples

- Q. Is it the case that if SP → ¬W is true then also ¬W → SP must be true?
- Sometimes a refutation is easy. Refutations are referred to as **counterexamples**.
- What do we have to do to refute in this case?
  - we have to find a model of SP→¬W in which ¬W → SP evaluates to false
  - there is such a model - [[W]] = F and [[SP]] = F
  - so the original statement is logically incorrect

Proofs

- Let us consider the propositional theory {SP → ¬W, SP}. Is it the case that in any model it can be the case that [[W]] = T?
- If the semantics of sentence is T in all models is there some way of verifying this **without manually checking all possible models**?
- Yes!
  - in logic, these are called formal proofs.
  - there are different proof systems, e.g. **natural deduction**
  - An example rule (modus ponens):

$$\frac{\varphi \qquad \varphi \rightarrow \psi}{\psi}$$

Propositional logic algorithmically

- Suppose I have a propositional theory. Can I write an algorithm to check that some formula ϕ is true in all models?

Relations and databases

- what is a database?

First name	Last name	Age	Salary
Bob	Walters	55	2000
Alice	Jones	36	3500
Lionel	Messi	36	1000000

Some notation

First name	Last name	Age	Salary
Bob	Walters	55	2000
Alice	Jones	36	3500
Lionel	Messi	36	1000000

- Let **String** be the datatype of strings
- Let **Nat** be the datatype of natural numbers (unsigned integers)
- Given two datatypes **T** and **U**, **T**×**U** is the datatype where values are pairs (t, u), where t is a value of **T** and u is a value of **U**
- Then our database is a collection of elements of **String**×**String**×**Nat**×**Nat**
- Mathematicians call such collections **relations**

Database queries and logic

First name	Last name	Age	Salary
Bob	Walters	55	2000
Alice	Jones	36	3500
Lionel	Messi	36	1000000

- There is a logic called **Regular Logic** that deals well with relations. It is closely related to the database query language SQL and languages such as datalog.
- Let us refer to a generic element of a relation as R(x, y, z, w).
  - R is called a **relation symbol** while x, y, z, w are **variables**
- The semantics is given by our database. So for example
  - [[R(“Bob”, ”Walters”, 55, 2000)]] = T
  - but [[R(“Bob”, ”Jones”, 55, 2000)]] = F

The existential quantifier

First name	Last name	Age	Salary
Bob	Walters	55	2000
Alice	Jones	36	3500
Lionel	Messi	36	1000000

- if ϕ is a formula then ∃x. ϕ is true exactly when there is some assignment to the variable x that makes ϕ true. In the formula ∃x. ϕ we say that x has been **quantified**.
- e.g.
  - [[∃x. R(x, “Walters, 55, 2000)]] = T
  - [[∃x. R(“Alice”, x, 55, 3500)]] = F

Free variables

First name	Last name	Age	Salary
Bob	Walters	55	2000
Alice	Jones	36	3500
Lionel	Messi	36	1000000

- If a variable is not quantified, it is said to be **free**
- for example ∃x.R(x,y,55,z) has two free variables: y and z.
- can we give a meaningful semantics to such formulas?
- Yes! [[∃x.R(x,y,55,z)]] = {(“Walters”,2000)}
  - Similarly [[∃x, y. R(x, y, 36, z)]] = {3500, 1000000}

Regular logic algorithmically

- **Q**. Suppose I have a regular theory. Can I write an algorithm to check that some formula ϕ is true in all models?
  - ie. if I know some facts about a database, can I figure out what is true about **all** databases in which the same facts hold?

- **A**. Yes! A very elegant algorithm was discovered in the 1970s.

Beyond regular logic

First name	Last name	Age	Salary
Bob	Walters	55	2000
Alice	Jones	36	3500
Lionel	Messi	36	1000000

- Adding negation gives something called **first order logic**.
- We can define the **universal quantifier** as a syntactic sugar.
- $\forall x.\ \phi = \neg\exists x.\neg\phi$. What is its semantics?
  - we can now write some more complex formulas, e.g.
    - $[\![\ \forall x, y, z.\ R(x,y,36,z) \rightarrow (z>3000)\ ]\!] = T$

First order logic, algorithmically

- Suppose I have a first order theory. Can I write an algorithm to check that some formula ϕ is true in all models?
  - there are formal proof systems for first order logic that allow you to prove all true formulas
  - that means that the problem is semi-decidable - if a formula is true in all models then an algorithm that constructs proofs will eventually discover it

First order logic on the natural numbers

- The early successes in logic convinced some mathematicians (famously David Hilbert) that all mathematics reduces to an algorithm
- Unfortunately Gödel ruined this dream
- It turns out that there is no complete proof system for first order logic over the natural numbers. This is the famous **Gödel incompleteness theorem**.
- For this reason, we cannot simply run an algorithm to check whether some famous theorems of mathematics are true.
- e.g. $\forall x.\exists y.\ (y \geq x) \wedge \text{prime}(y) \wedge \text{prime}(y+2)$ - the **twin primes conjecture**

Logic in computer science - the big picture

- We have only touched on the relevance of logic in computer science
  - logic is used to verify the correctness of programs - **model checking**
  - logic is used as a foundation for type theory and programming languages - especially **functional programming**
  - logic is used as a basis for understanding challenging programming paradigms like **concurrent programming** and **quantum programming**
  - …